

EC 553



Communication Networks

Mohamed Khedr

<http://webmail.aast.edu/~khedr>

Syllabus

- Tentatively

Week 1	Overview
Week 2	Packet Switching
Week 3	IP addressing and subnetting
Week 4	IP addressing and subnetting
Week 5	Introduction to Routing concept, Routing algorithms
Week 6	Routing protocols
Week 7	Multiple Access I
Week 8	Multiple access II
Week 9	LAN networks
Week 10	Token ring networks
Week 11	VOIP
Week 12	WLAN
Week 13	TCP
Week 14	Congestion control
Week 15	QOS

Dijkstra's Algorithm

- finds shortest paths from given source node s to all other nodes
- by developing paths in order of increasing path length
- algorithm runs in stages (next slide)
 - each time adding node with next shortest path
- algorithm terminates when all nodes processed by algorithm (in set T)

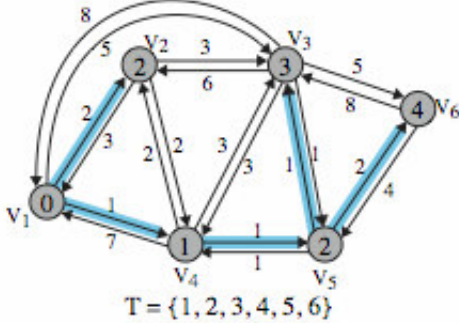
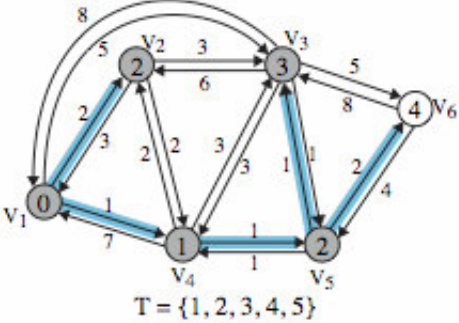
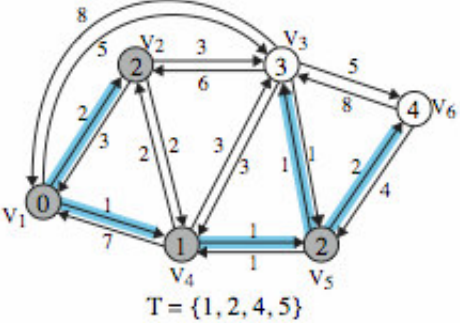
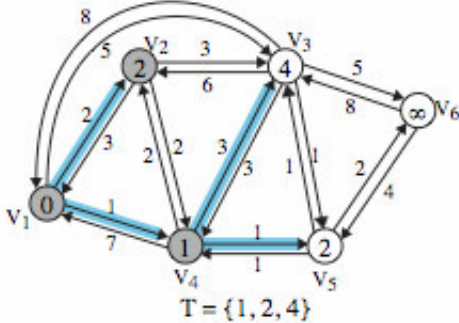
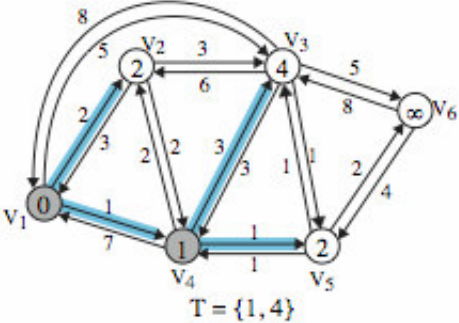
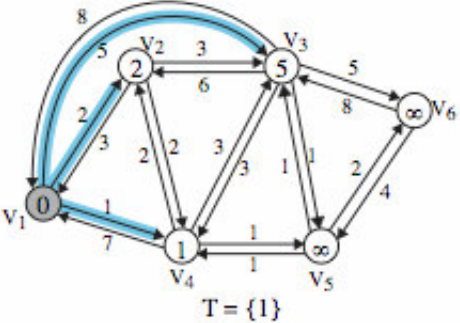
Dijkstra's Algorithm Method

- Step 1 [Initialization]
 - $T = \{s\}$ Set of nodes so far incorporated
 - $L(n) = w(s, n)$ for $n \neq s$
 - initial path costs to neighboring nodes are simply link costs
- Step 2 [Get Next Node]
 - find neighboring node not in T with least-cost path from s
 - incorporate node into T
 - also incorporate the edge that is incident on that node and a node in T that contributes to the path
- Step 3 [Update Least-Cost Paths]
 - $L(n) = \min[L(n), L(x) + w(x, n)]$ for all $n \notin T$
 - if latter term is minimum, path from s to n is path from s to x concatenated with edge from x to n

Dijkstra's Algorithm Example

Iter	T	L(2)	Path	L(3)	Path	L(4)	Path	L(5)	Path	L(6)	Path
1	{1}	2	1-2	5	1-3	1	1-4	∞	-	∞	-
2	{1,4}	2	1-2	4	1-4-3	1	1-4	2	1-4-5	∞	-
3	{1, 2, 4}	2	1-2	4	1-4-3	1	1-4	2	1-4-5	∞	-
4	{1, 2, 4, 5}	2	1-2	3	1-4-5-3	1	1-4	2	1-4-5	4	1-4-5-6
5	{1, 2, 3, 4, 5}	2	1-2	3	1-4-5-3	1	1-4	2	1-4-5	4	1-4-5-6
6	{1, 2, 3, 4, 5, 6}	2	1-2	3	1-4-5-3	1	1-4	2	1-4-5	4	1-4-5-6

Dijkstra's Algorithm Example



Bellman-Ford Algorithm

- find shortest paths from given node subject to constraint that paths contain at most one link
- find the shortest paths with a constraint of paths of at most two links
- and so on

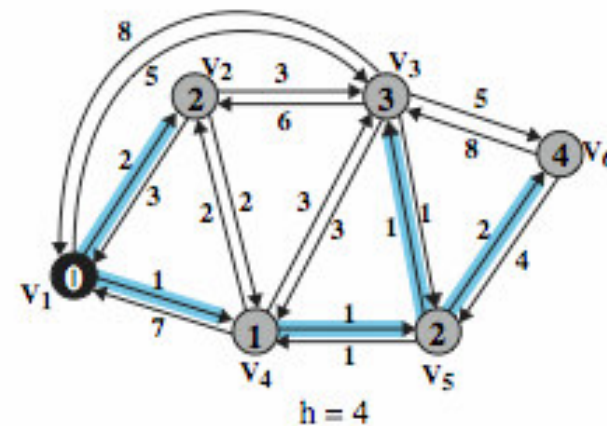
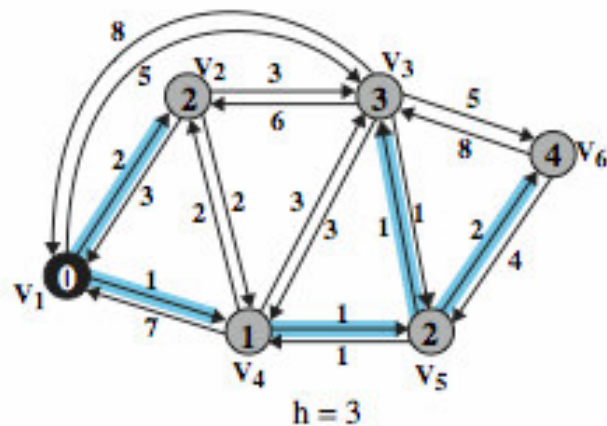
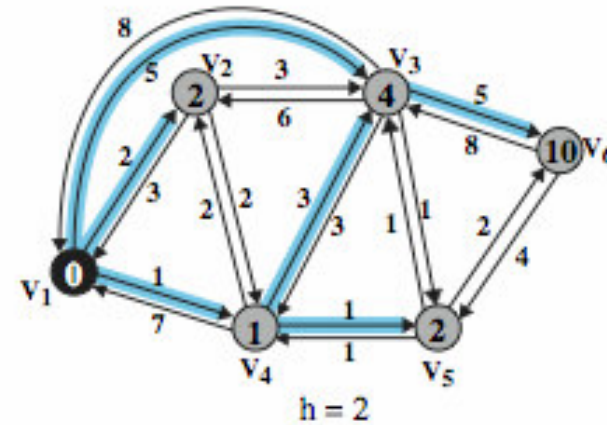
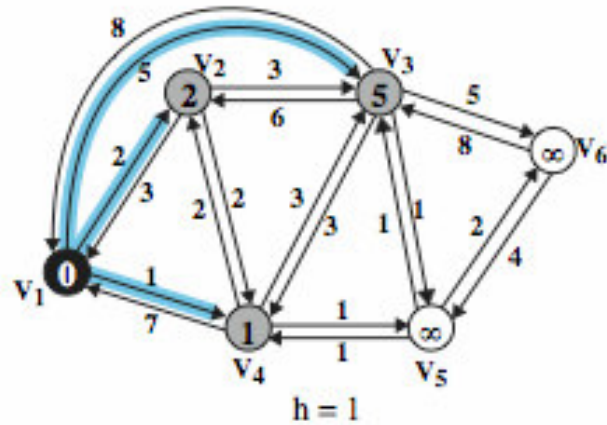
Bellman-Ford Algorithm

- step 1 [Initialization]
 - $L_0(n) = \infty$, for all $n \neq s$
 - $L_h(s) = 0$, for all h
- step 2 [Update]
 - for each successive $h \geq 0$
 - for each $n \neq s$, compute:
$$L_{h+1}(n) = \min_j [L_h(j) + w(j, n)]$$
 - connect n with predecessor node j that gives min
 - eliminate any connection of n with different predecessor node formed during an earlier iteration
 - path from s to n terminates with link from j to n

Results of Bellman-Ford Example

h	$L_h(2)$	Path	$L_h(3)$	Path	$L_h(4)$	Path	$L_h(5)$	Path	$L_h(6)$	Path
0	∞	-	∞	-	∞	-	∞	-	∞	-
1	2	1-2	5	1-3	1	1-4	∞	-	∞	-
2	2	1-2	4	1-4-3	1	1-4	2	1-4-5	10	1-3-6
3	2	1-2	3	1-4-5-3	1	1-4	2	1-4-5	4	1-4-5-6
4	2	1-2	3	1-4-5-3	1	1-4	2	1-4-5	4	1-4-5-6

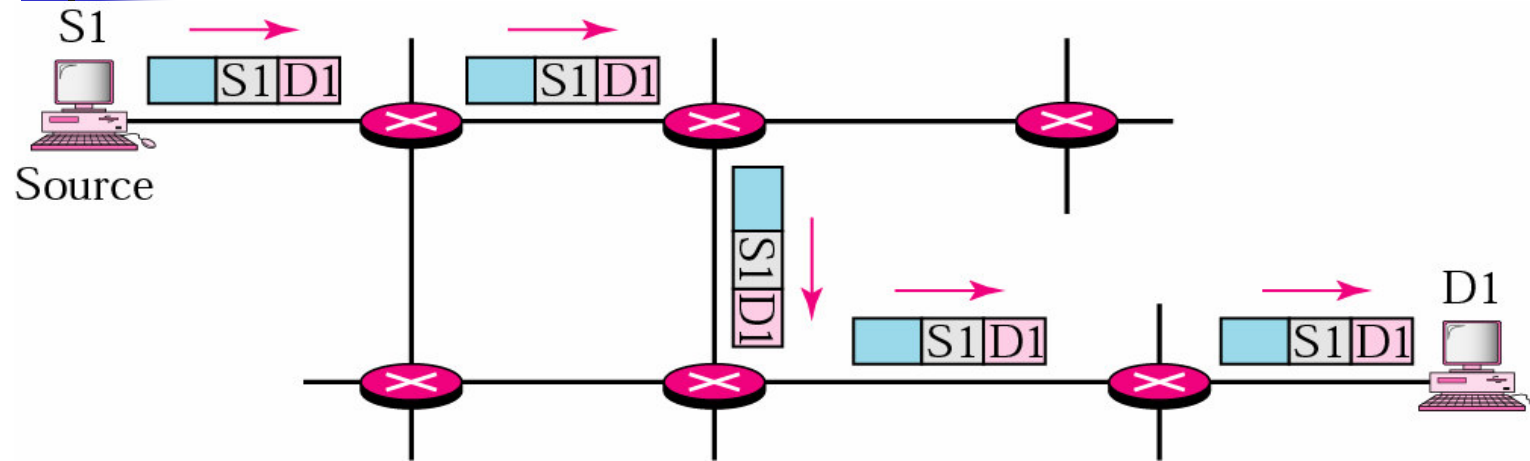
Example of Bellman-Ford Algorithm



Chapter 21

Unicast and Multicast Routing: Routing Protocols

Figure 21.1 Unicasting



In unicast routing, the router forwards the received packet through only one of its ports.

21.2 Unicast Routing Protocols

RIP

OSPF

BGP

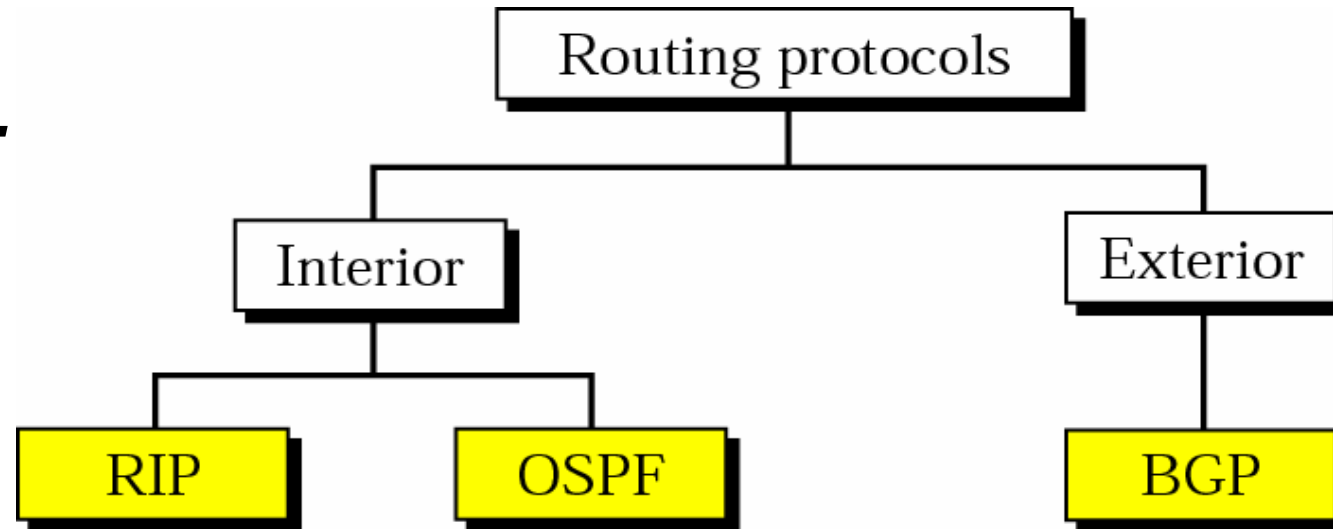
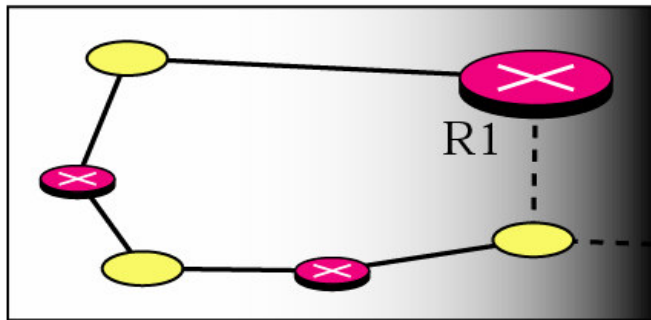


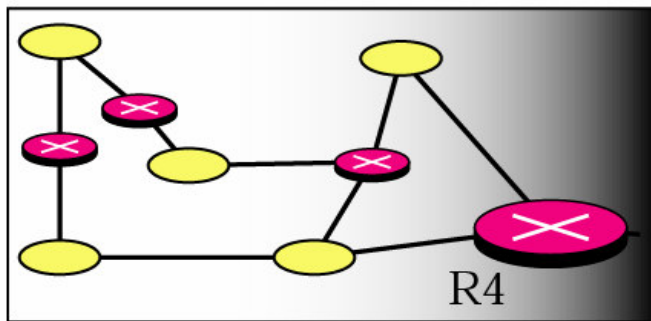
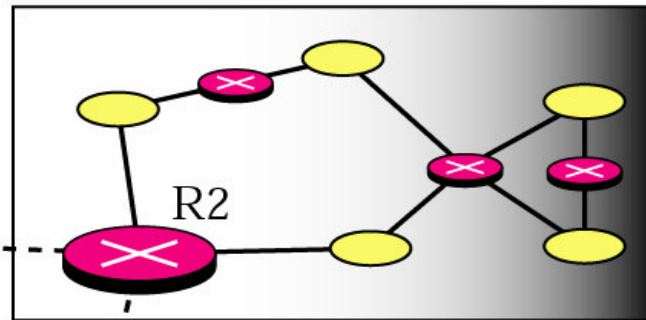
Figure 21.3 Autonomous systems

AS is a group of networks and routers under the authority of a single administration.

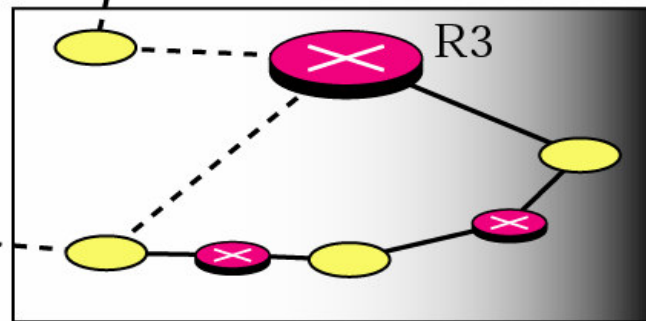
Autonomous system



Autonomous system



Autonomous system



Autonomous system

R1, R2, R3, R4 use and interior and exterior routing protocols.

Other routers use interior only

Solid lines for interior protocols

Dashed lines for exterior protocols

Approaches to Routing – Distance-vector

- each node (router or host) exchange information with neighboring nodes
- first generation routing algorithm for ARPANET
 - eg. used by Routing Information Protocol (RIP)
- each node maintains vector of link costs for each directly attached network and distance and next-hop vectors for each destination
- requires transmission of much info by routers
 - distance vector & estimated path costs
- changes take long time to propagate
- Based on Bellman-Ford algorithm

Distance Vector

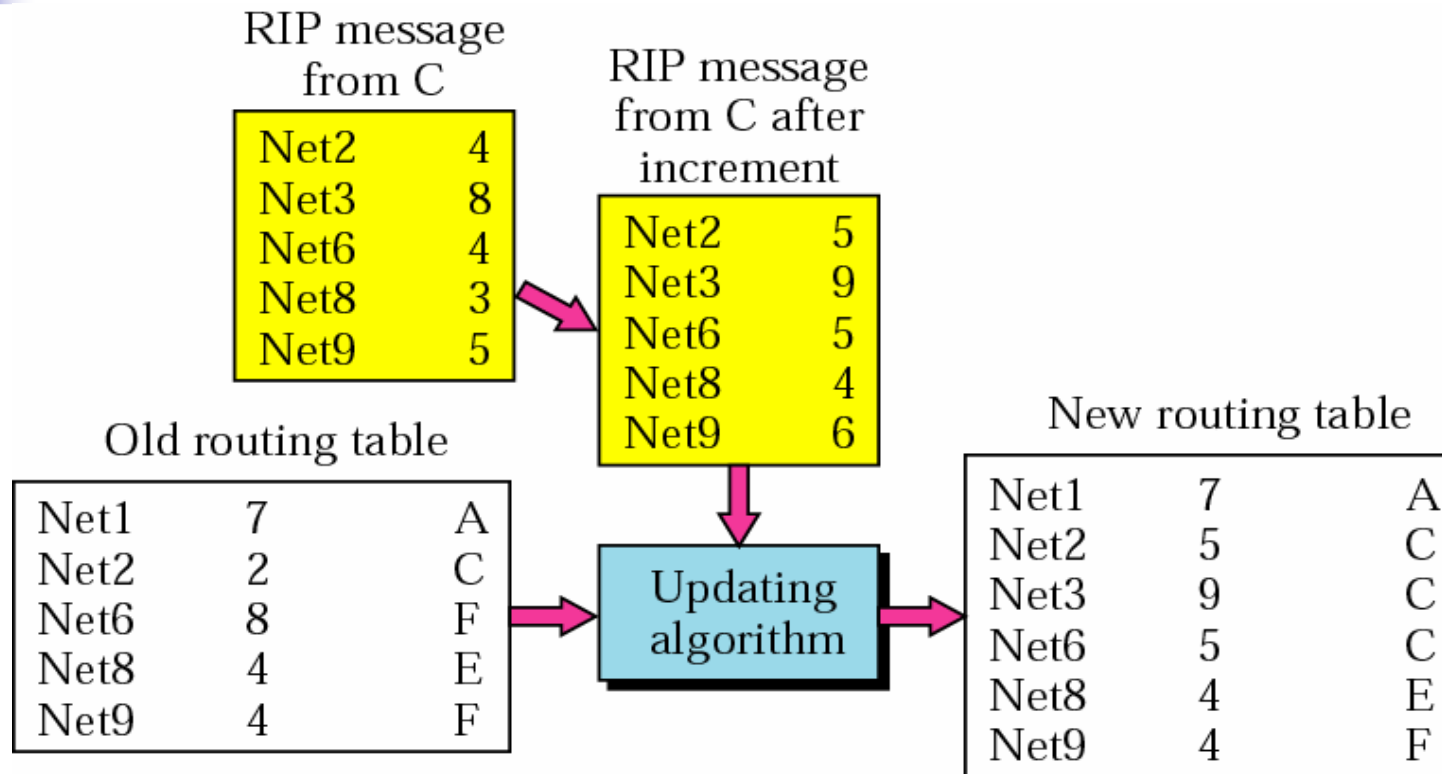
- Sharing knowledge about the entire AS
- Sharing only with neighbors
- Sharing at regular intervals

RIP Updating Algorithm

Receive: a response RIP message

- 1. Add one hop to the hop count for each advertised destination.**
- 2. Repeat the following steps for each advertised destination:**
 - 1. If (destination not in the routing table)**
 - 1. Add the advertised information to the table.**
 - 2. Else**
 - 1. If (next-hop field is the same)**
 - 1. Replace entry in the table with the advertised one.**
 - 2. Else**
 - 1. If (advertised hop count smaller than one in the table)**
 - 1. Replace entry in the routing table.**
 - 3. Return.**

Figure 21.4 Example of updating a routing table



- Net1: No news, do not change
- Net2: Same next hop, replace
- Net3: A new router, add
- Net6: Different next hop, new hop count smaller, replace
- Net8: Different next hop, new hop count the same, do not change
- Net9: Different next hop, new hop count larger, do not change

Figure 21.5 Initial routing tables in a small autonomous system

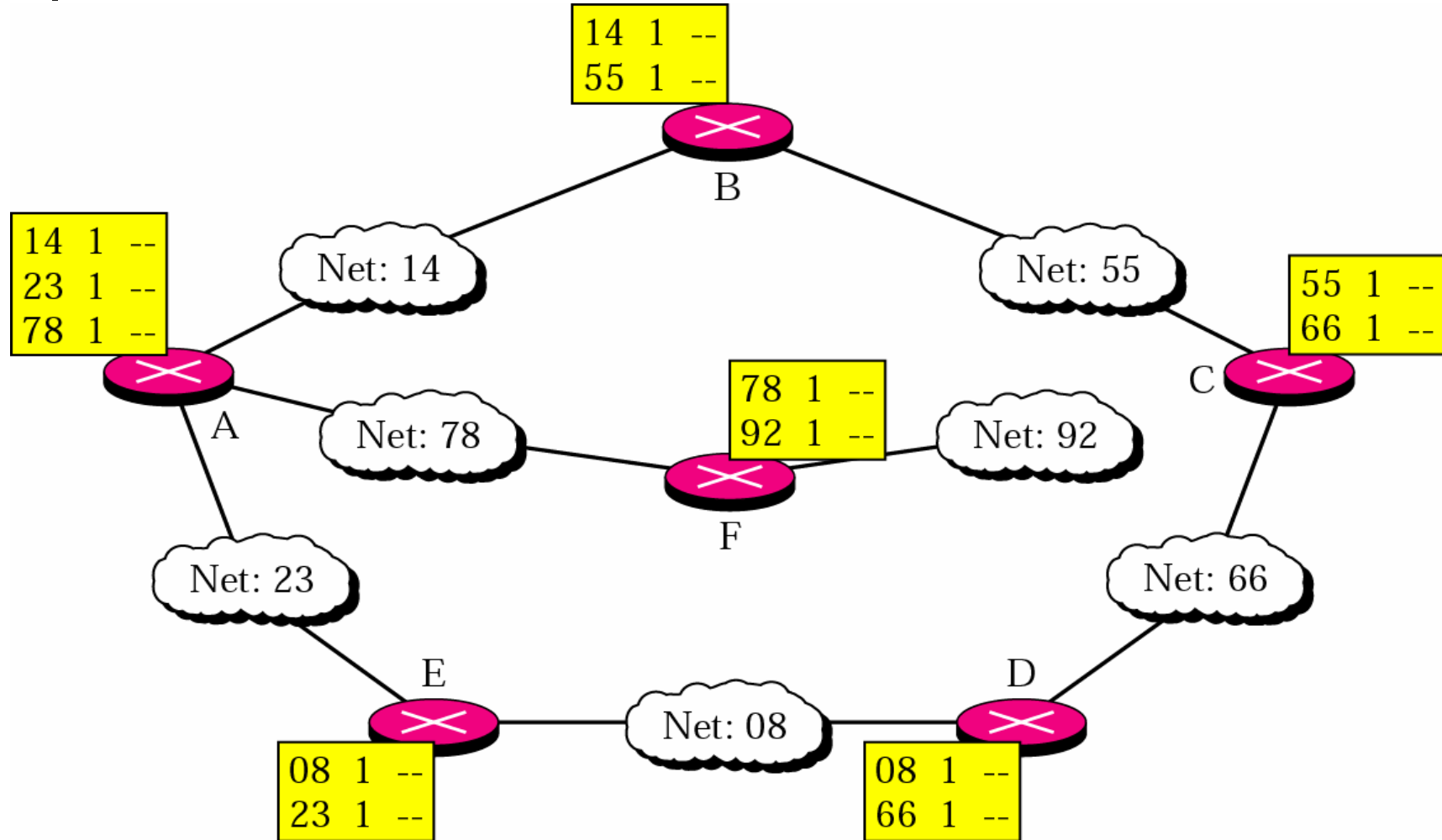


Figure 21.6 Final routing tables for Figure 21.5

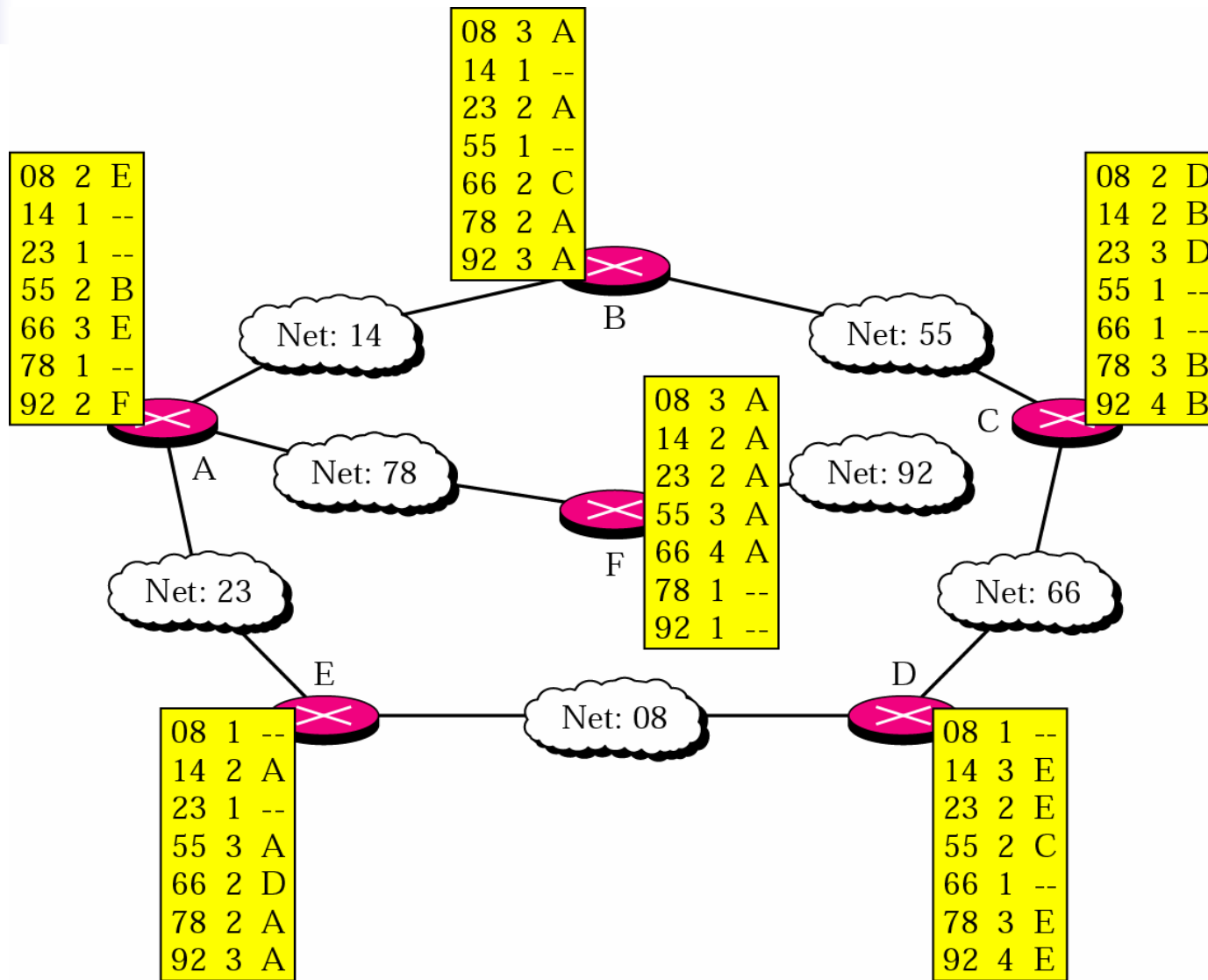
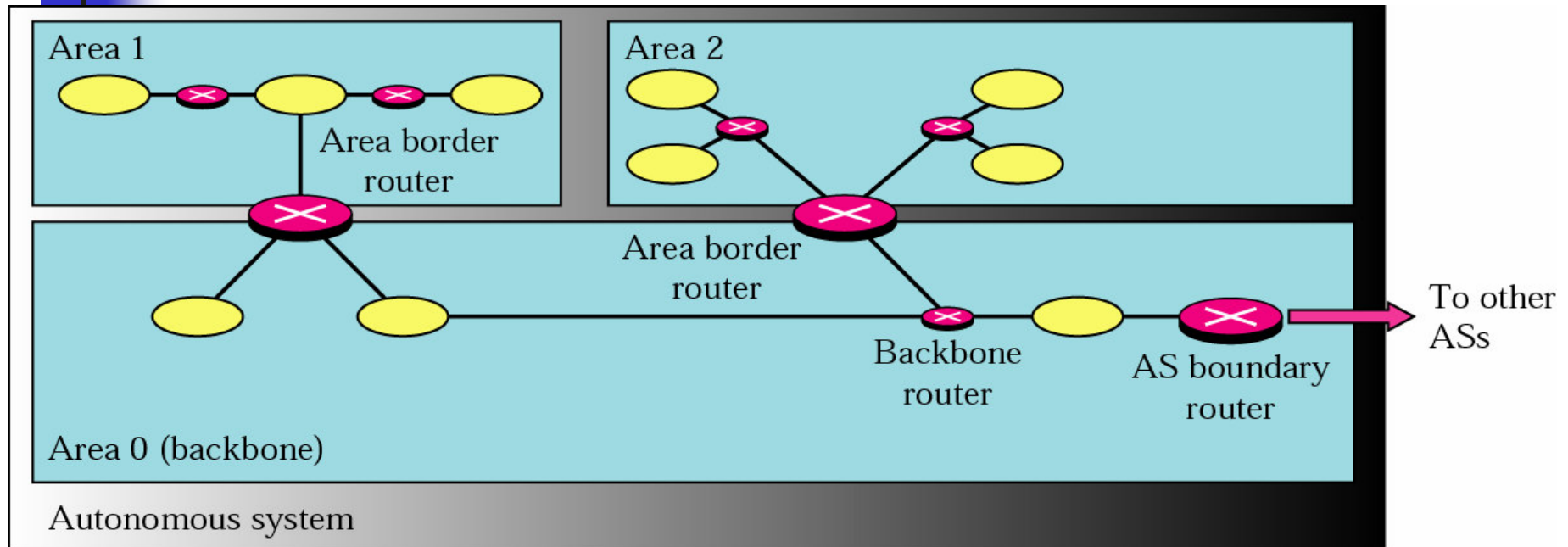


Figure 21.7 Areas in an autonomous system



- Area: collection of hosts, routers and networks contained inside an AS.
- Routers inside an area floods the area with routing information
- Routers at the boarder summarize information about area and send it to other areas.
- Backbone area is the area where all other areas must be connected.
- Routers inside the backbone area are called backbone routers

Approaches to Routing – Link-state

- designed to overcome drawbacks of distance-vector
- each router determines link cost on each interface
- advertises set of link costs to all other routers in topology
- if link costs change, router advertises new values
- each router constructs topology of entire configuration
 - can calculate shortest path to each dest
 - use to construct routing table with first hop to each dest
- do not use distributed routing algorithm, but any suitable alg to determine shortest paths, eg. Dijkstra's algorithm
- Open Shortest Path First (OSPF) is a link-state protocol

Link state

- Sharing knowledge about the neighbors
- Sharing with every other router
- Sharing when there is a change

Open Shortest Path First (RFC2328)

- IGP of Internet
- replaced Routing Information Protocol (RIP)
- uses Link State Routing Algorithm
 - each router keeps list of state of local links to network
 - transmits update state info
 - little traffic as messages are small and not sent often
- uses least cost based on user cost metric
- topology stored as directed graph
 - vertices or nodes (router, transit or stub network)
 - edges (between routers or router to network)